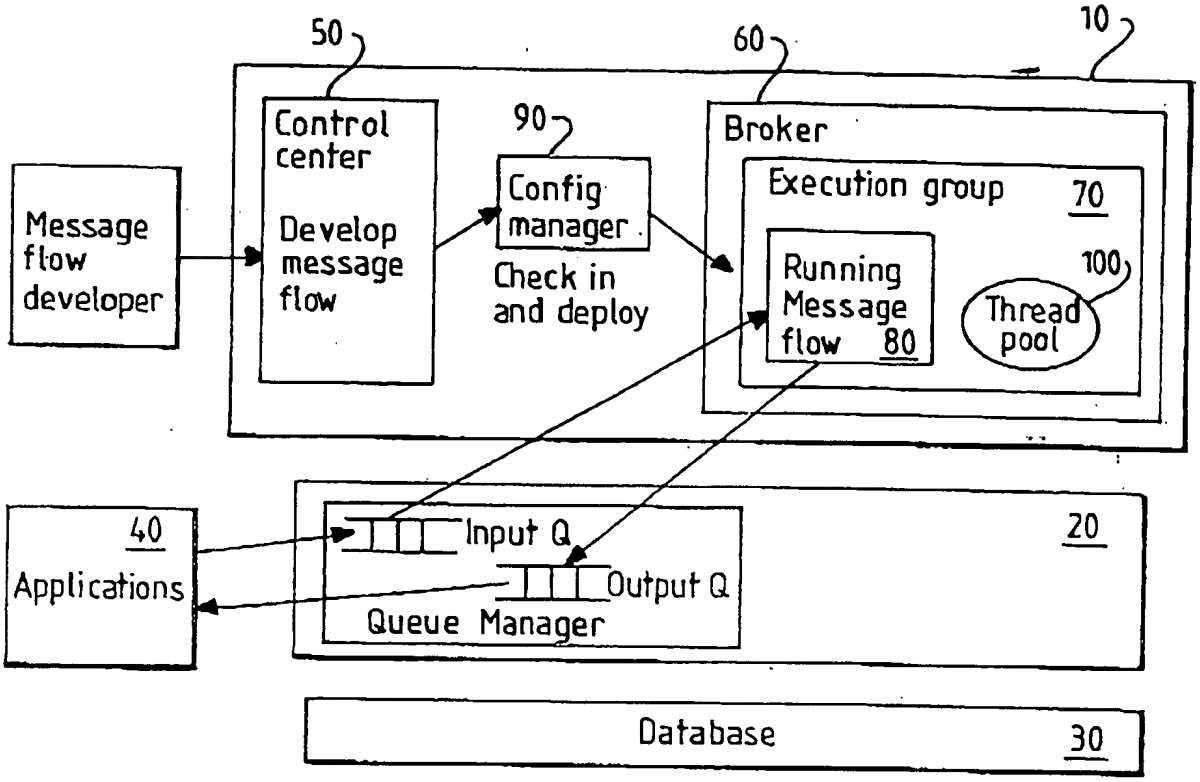
50
60
10
Control center
Develop message flow
90
Config manager
Check in and deploy
Broker
Execution group
70
Running Message flow 80
100
Thread pool
Message flow developer
40
Applications
Input Q
Output Q
Queue Manager
20
Database
30

FIG. 1

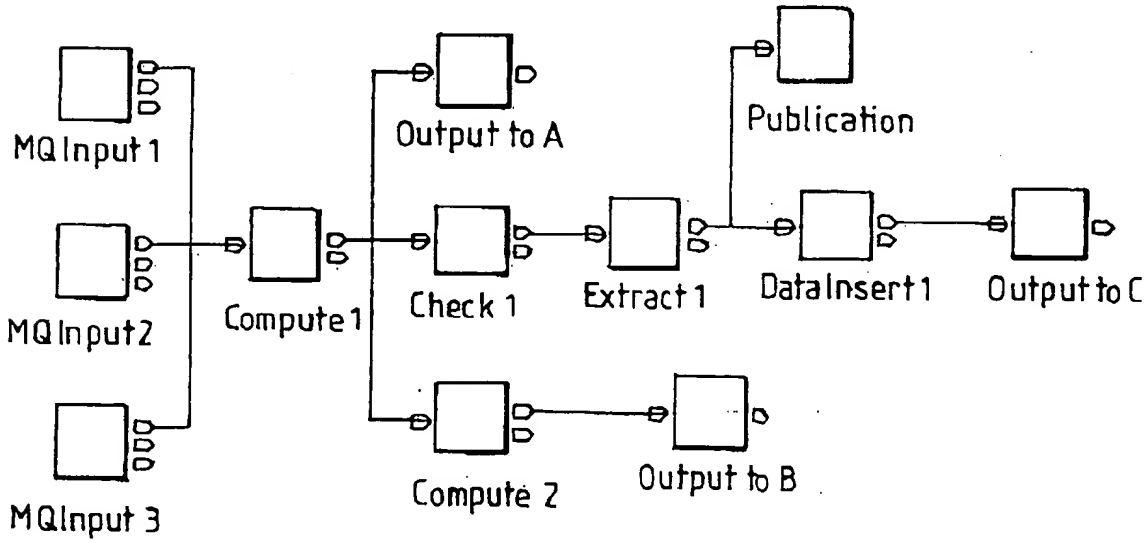
MQ Input 1
MQ Input 2
MQ Input 3
Compute 1
Output to A
Check 1
Compute 2
Extract 1
Publication
DataInsert 1
Output to C
Output to B

FIG. 2

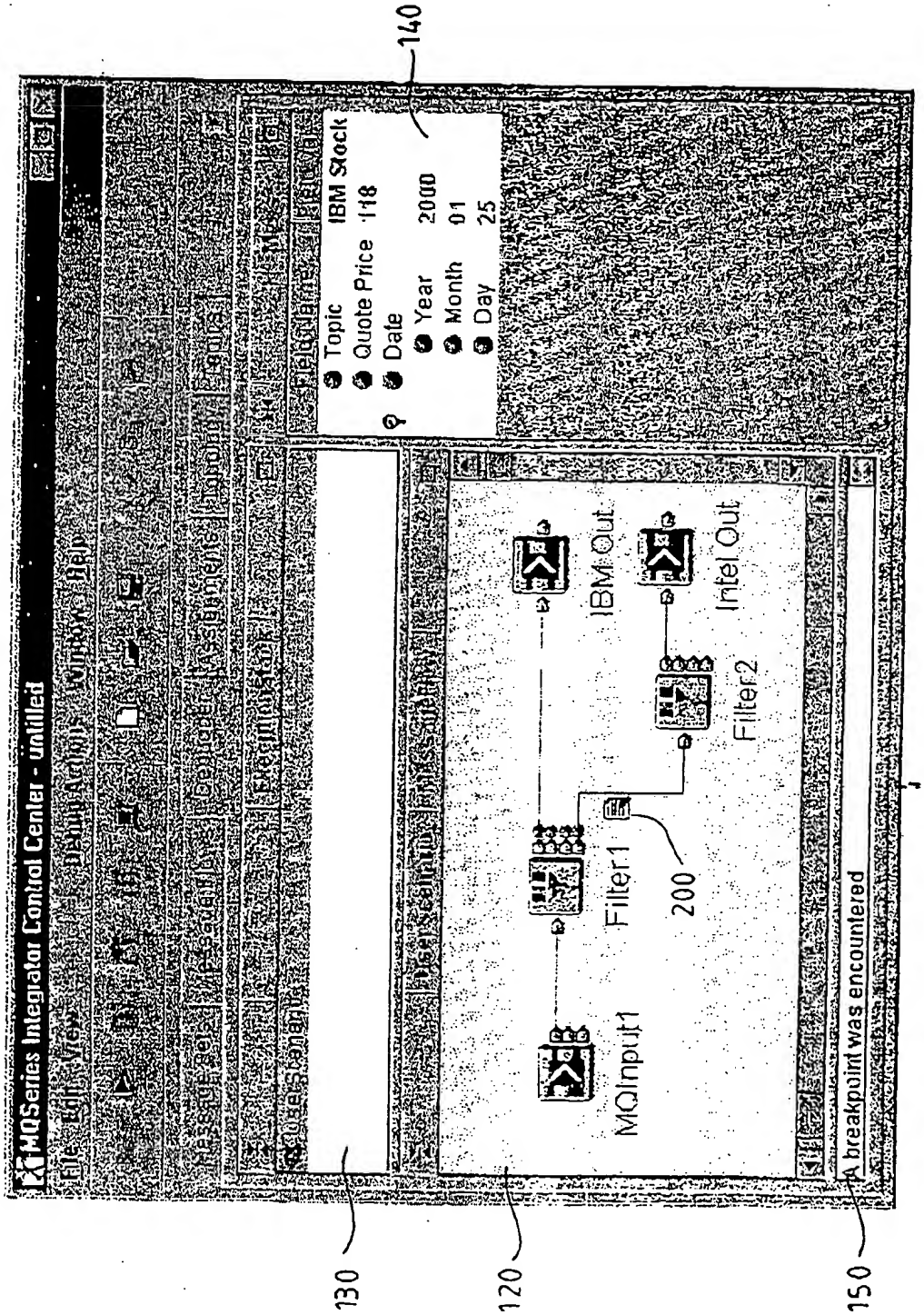
MQSeries Integrator Control Center - Untitled
File Edit View Deploy Actions Window Help
Topic IBM Stock
Quote Price 118
Date
Year 2000
Month 01
Day 25
140
MQInput1
Filter1
200
IBM Out
Filter2
Intel Out
A breakpoint was encountered
130
120
150

FIG. 3

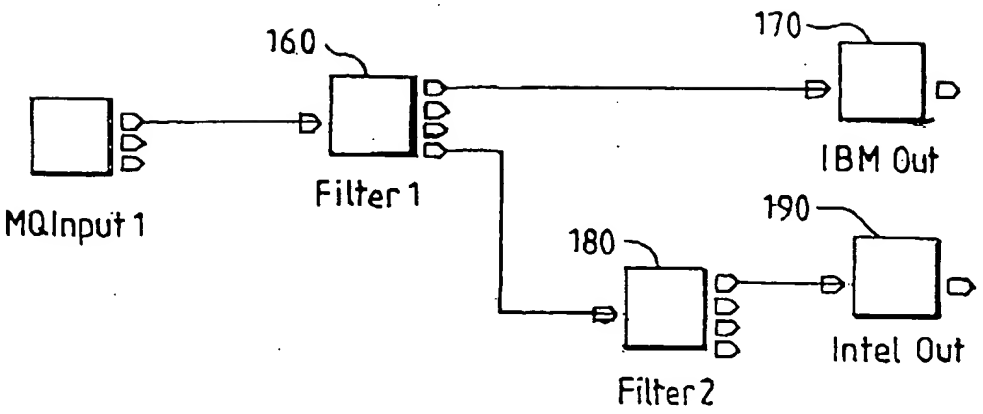
160
170
MQInput 1
Filter 1
IBM Out
180
190
Filter 2
Intel Out

FIG. 4

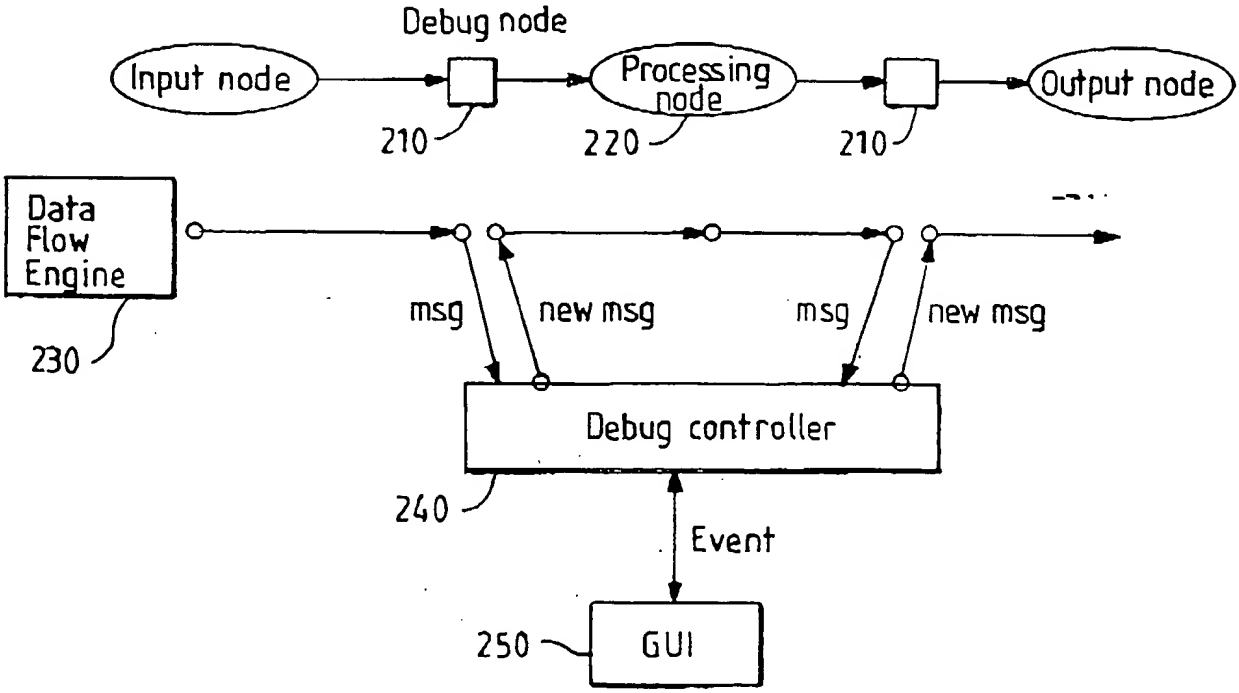
Debug node
Input node
Processing node
Output node
210
220
210
Data Flow Engine
230
msg
new msg
msg
new msg
Debug controller
240
Event
250
GUI

FIG. 5

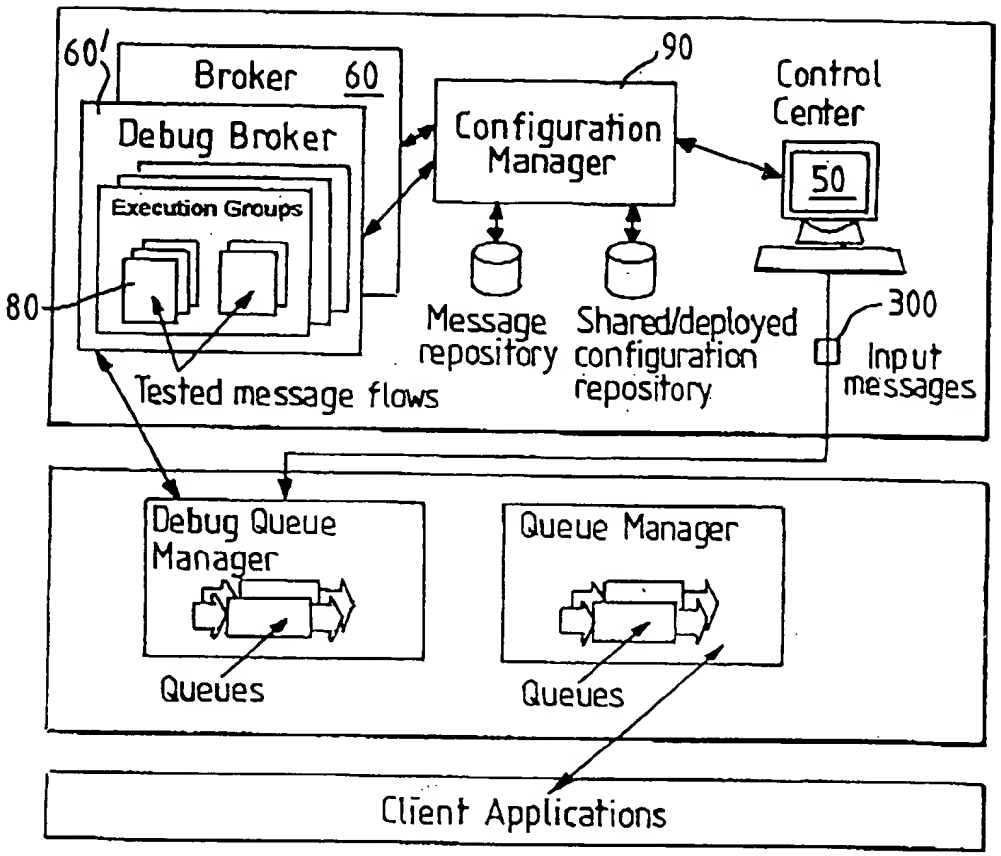
60'
Broker 60
Debug Broker
Execution Groups
80
Tested message flows
90
Configuration Manager
Control Center
50
Message repository
Shared/deployed configuration repository
300
Input messages
Debug Queue Manager
Queues
Queue Manager
Queues
Client Applications

FIG. 6

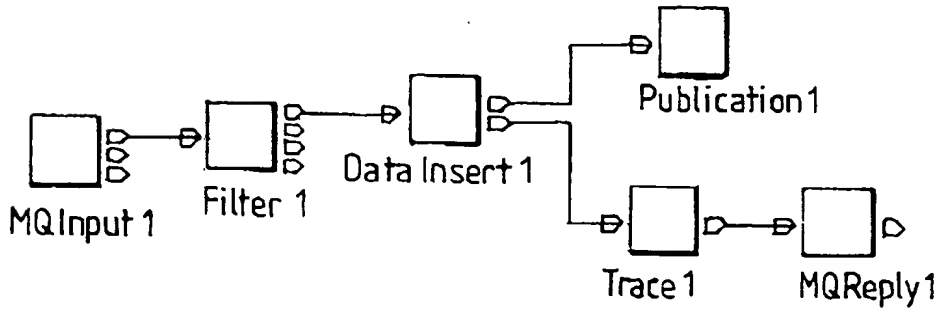
MQInput 1
Filter 1
Data Insert 1
Publication 1
Trace 1
MQReply 1

FIG. 7

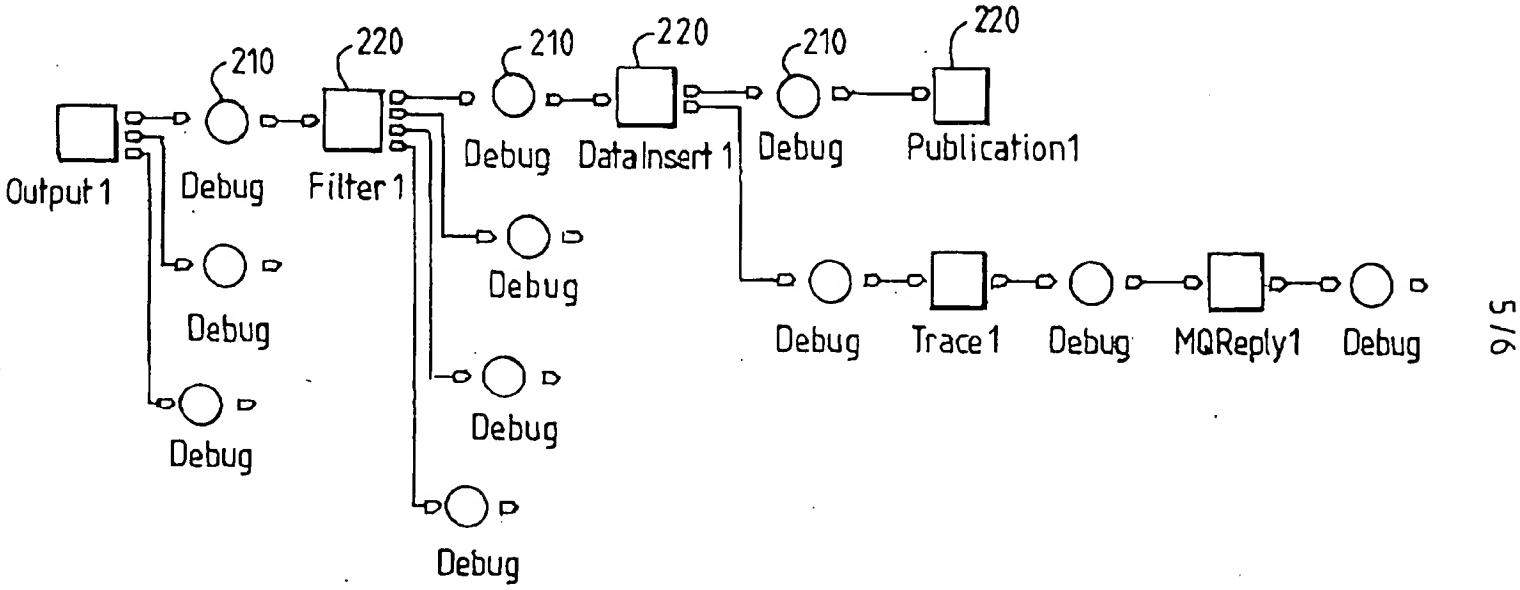
210
220
210
220
210
220
Output 1
Debug
Filter 1
Debug
DataInsert 1
Debug
Publication1
Debug
Debug
Debug
Debug
Debug
Debug
Trace 1
Debug
MQReply1
Debug

FIG. 8

Debugger Algorithm

- Upon receiving communication from a debug node:
  - Receive debug node identification
  - Receive message structure and content
  - Is there a breakpoint associated with this position?
    - Yes: display message and position indication and wait for user command.
  - Send message structure and content back to debug node
  - Send resume indication to debug node

FIG. 9

Debug Node Algorithm

- Receive message from input port
- Initiate communication with debugger
- Send identification of the debug node
- Send message structure and content
- Wait for indication to resume (from the debugger)
- Receive message structure and content from debugger
- Conclude communication with debugger
- Propagate the message (received from the debugger) to the out port

FIG. 10

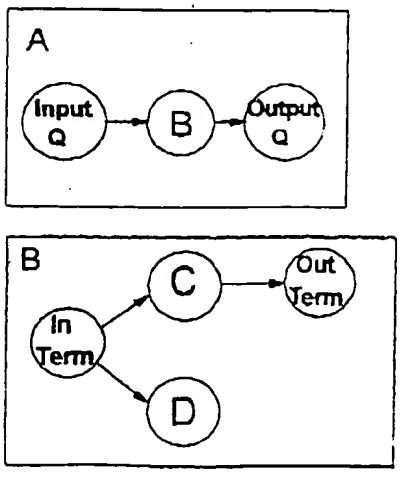

FIG. 11